IN THE CLAIMS

Please amend claims 1, 4, and 7 as indicated below.

1. (Currently Amended) A method comprising:

receiving an interrupt from an IO device;

converting said interrupt into an upstream memory write interrupt by generating a memory write request to a predetermined address of a memory space, the memory write request being processed via one or more memory cycles; and

converting said upstream memory write interrupt into a front side bus (FSB) interrupt transaction, wherein one or more processors coupled to the FSB are capable of receiving the FSB interrupt as a part of a FSB transaction.

2. (Original) The method as in claim 1, wherein said interrupt is generated by a peripheral component interconnect (PCI) device.

3. (Original) The method as in claim 2, wherein said FSB interrupt is received by a processor.

4. (Currently Amended) A method comprising:

receiving a message signaled interrupt (MSI) interrupt;

forwarding said MSI interrupt to a memory controller via a memory write request addressed to a predetermined address of a memory space, the memory write request being processed via one or more memory cycles; and

converting said MSI interrupt into an FSB interrupt transaction, wherein one or more processors coupled to the FSB are capable of receiving the FSB interrupt as a part of a FSB transaction.

5. (Original) The method as in claim 4, wherein said MSI interrupt is generated by a PCI device, and wherein said FSB interrupt is received by a processor.

6. (Original) The method as in claim 5, wherein said FSB interrupt is received by a processor.

7. (Currently Amended) A method comprising:

receiving a hardware signal, the hardware signal being generated from a hardware interrupt;

converting said hardware signal into an upstream memory write interrupt via a memory write request addressed to a predetermined address of a memory space, the memory write request being processed via one or more memory cycles; and

converting said upstream memory write interrupt into an FSB interrupt transaction, wherein one or more processors coupled to the FSB are capable of receiving the FSB interrupt as a part of a FSB transaction.

8. (Original) The method as in claim 7, wherein said hardware signal is generated by a PCI device, and wherein said FSB interrupt is received by a processor.

9. (Original) The method as in claim 8, wherein said FSB interrupt is received by a processor.

10. (Previously Presented) An apparatus comprising:

a chipset, configured to receive an interrupt from at least one I/O (input and output) device, to convert said interrupt into an upstream memory write interrupt processed as one or more memory cycles, and to convert said upstream memory write interrupt into an FSB interrupt transaction, which is processed on a FSB (front side bus).

11. (Original) The apparatus as in claim 10, said chipset further comprising at least one of an I/O controller hub (ICH), P64H, AGP device.

12. (Previously Presented) The apparatus as in claim 11, further comprising an I/O component of an advanced programmable interrupt controller (IOxAPIC) configured to convert said interrupt into said upstream memory write interrupt.

13. (Previously Presented) The apparatus as in claim 12, said chipset further comprising a HUB interface coupled on a first end to said IOxAPIC and coupled on a second end to a MCH, wherein said memory controller hub (MCH) configured to convert said upstream memory write interrupt into said FSB interrupt transaction.

14. (Original) The apparatus as in claim 10, wherein said interrupt is generated by a PCI device, and wherein said chipset is coupled to a processor.

15. (Original) The apparatus as in claim 10, wherein said interrupt is a MSI interrupt.

16. (Original) The apparatus as in claim 15, said chipset further comprising a HUB interface coupled on a first end to an ICH and coupled on a second end to a MCH, wherein said ICH configured to forward said MSI interrupt to said HUB interface, and wherein said MCH configured to convert said MSI interrupt into a FSB interrupt transaction.

17. (Original) The apparatus as in claim 16, wherein said MCH configured to ensure at least one data pipe of a HUB interface is flushed upstream before propagating an interrupt upstream.

18. (Original) The apparatus as in claim 16, wherein said MCH configured to receive an end of interrupt (EOI) from a processor and broadcast said EOI to at least one downstream HUB interface IOxAPIC generating at least one level mode interrupt.

19. – 30. (Canceled).

31. (Previously Presented) A method, comprising:

receiving, at an IO (input and output) controller, an interrupt request from an IO device;

generating, at the IO controller, a memory request at a predetermined address in response to the interrupt request; and

transmitting the memory request to a memory controller, the memory request being processed at the memory controller as one or more memory cycles, which are routed to a bus coupling with one or more processors as a part of one or more interrupt message transactions on the bus.

32. (Previously Presented) The method of claim 31, wherein the memory request is a memory write request to the memory controller.

33. (Previously Presented) The method of claim 31, wherein the interrupt request is one of interrupt request hardware signal and a memory write request to the IO controller.

34. (Previously Presented) An apparatus, comprising:

a first interface to receive at least one interrupt request from at least one IO device;

a circuit coupled to the first interface to generate a memory request at a predetermined address in response to the at least one interrupt request; and

a second interface coupled to the circuit to transmit the memory request to a memory controller, the memory request being processed as one or more memory cycles on a bus coupling with one or more processors.

35. (Previously Presented) The apparatus of claim 34, wherein the memory request is a memory write request to the memory controller.

36. (Previously Presented) The apparatus of claim 34, wherein the interrupt request is one of interrupt request hardware signal and a memory write request to the IO controller.

37. (Previously Presented) A method, comprising:

receiving, at a memory controller, a memory request addressed to a predetermined address; and

generating one or more memory cycles on a bus coupling to one or more processors in response to the memory request, the one or more memory cycles being processed by at least one of the one or more processors as a part of one or more interrupt message transactions.

38. (Previously Presented) The method of claim 37, wherein the memory request is received from an IO (input and output) controller based on an interrupt request of one or more IO devices coupled to the IO controller.

39. (Previously Presented) The method of claim 37, wherein the memory request is received from one or more IO devices coupled to the memory controller.

40. (Previously Presented) The method of claim 37, further comprising redirecting the one or more interrupt message transactions to a processor based on task priority information associated with the processor.

41. (Previously Presented) The method of claim 40, further comprising substituting one or more bits of an address field of the memory request with information related to the task priority information prior to the redirecting.

42. (Previously Presented) The method of claim 40, further comprising periodically receiving the task priority information from the processor.

43. (Previously Presented) The method of claim 40, wherein the processor has a lowest priority among the one or more processors.

44. (Previously Presented) An apparatus, comprising:

a first interface to receive a memory request from an IO (input and output) controller, the memory request being addressed to a predetermined address;

a second interface capable of coupling to a bus coupling with one or more processors; and

a circuit coupled to the first and second interfaces to generate one or more memory cycles on the bus in response to the memory request, the one or more memory cycles being processed by at least one of the one or more processors as a part of one or more interrupt message transactions.

45. (Previously Presented) The apparatus of claim 44, wherein the memory request is received from an IO (input and output) controller based on an interrupt request of one or more IO devices coupled to the IO controller.

46. (Previously Presented) The apparatus of claim 44, wherein the memory request is received from one or more IO devices coupled to the memory controller.

47. (Previously Presented) The apparatus of claim 44, further comprising a redirection logic to redirect the one or more interrupt message transactions to a processor based on task priority information associated with the processor.

48. (Previously Presented) The apparatus of claim 47, wherein the redirection logic substitutes one or more bits of an address field of the memory request with information related to the task priority information prior to the redirecting.

49. (Previously Presented) The apparatus of claim 47, further comprising one or more task priority registers (TPRs) to store the task priority information.

50. (Previously Presented) The apparatus of claim 49, wherein the TPRS periodically receives the task priority information from the processor.

51. (Previously Presented) An apparatus, comprising:

an IO (input and output) controller capable of receiving an interrupt request from one or more IO devices, the IO controller generating a memory request addressed to a predetermined address in response to the interrupt request; and

a memory controller coupled to the IO controller to generate one or more memory cycles on a bus coupling to one or more processors in response to the memory request, the one or more memory cycles being processed by at least one of the one or more processors as a part of one or more interrupt message transactions.

52. (Previously Presented) The apparatus of claim 51, wherein the memory request is a memory write request to the memory controller.

53. (Previously Presented) The apparatus of claim 51, wherein the interrupt request is one of interrupt request hardware signal and a memory write request to the IO controller.

54. (Previously Presented) The apparatus of claim 51, wherein the memory controller further receives one or more memory requests addressed to the predetermined address directly from the one or more IO devices.

55. (Previously Presented) The apparatus of claim 51, further comprising a redirection logic to redirect the one or more interrupt message transactions to a processor based on task priority information associated with the processor.

56. (Previously Presented) The apparatus of claim 55, wherein the redirection logic substitutes one or more bits of an address field of the memory request with information related to the task priority information prior to the redirecting.

57. (Previously Presented) The apparatus of claim 55, further comprising one or more task priority registers (TPRs) to store the task priority information.

58. (Previously Presented) The apparatus of claim 57, wherein the TPRS periodically receives the task priority information from the processor.

59. (Previously Presented) A system, comprising:

a bus;

one or more processors coupled to the bus;

an IO (input and output) controller capable of receiving an interrupt request from one or more IO devices, the IO controller generating a memory request addressed to a predetermined address in response to the interrupt request; and

a memory controller coupled to the IO controller and the bus to generate one or more memory cycles on the bus in response to the memory request, the one or more memory cycles

being processed by at least one of the one or more processors as a part of one or more interrupt message transactions.

60. (Previously Presented) The system of claim 59, wherein the memory controller comprises:

a redirect logic to redirect one or more interrupts generated from one processor to another processor based on task priority information; and

one or more task priority registers (TPRs) to store and to receive the task priority information from at least one of the one or more processors.